Daniela Kacso
Christian Brandl

Bochum, den 28. 04. 2008

Übungen zur Vorlesung

Effiziente Algorithmen

SS 08

Blatt 3

Hinweis:
Auf der Homepage sind wichtige Informationen zu der Klausur und der Wiederholungsklausur zu finden!

Aufgabe 3.1

Off-line Berechnung des Kürzesten-Pfade-Baums:
Beschreibe einen Algorithmus, der als Eingabe das Array $d(v), v = 0, \ldots, n-1$ der kürzesten Pfadlängen erhält und die Pfad-Pointer berechnet.

Aufgabe 3.2

Wende die FIFO-Implementierung des generischen Kürzeste-Pfade Algorithmus auf den folgenden Graphen an. Gib dazu für jede Iteration der while-Schleife an, wie sich LIST, $d(j)$ und $F(j)$ ändern.
Bei Ausführung des Algorithmus wende folgende Zusatzregel an: Durchlaufe die Adjazenzliste eines Knotens in aufsteigender Reihenfolge der Knotenindizes!

$A_s : (1, 5), (2, 10)$ $\quad$ $A_4 : (5, 5), (6, -5)$
$A_1 : (2, 10), (5, 15)$ $\quad$ $A_5 : (6, 10)$
$A_2 : (3, 10)$ $\quad$ $A_6 : (2, -12)$
$A_3 : (1, -10), (4, 10)$

Bei den Adjazenzlisten steht der erste Eintrag für den adjazenten Knoten und der zweite Eintrag für die Kosten der entsprechenden Kante.

Aufgabe 3.3

Zeige, dass der generische Kürzeste-Pfade Algorithmus A_{gen} auch für den Fall terminiert, dass der Graph beliebige reelle Kantenkosten besitzt und keine negativen Kreise enthält.

Aufgabe 3.4

Die Laufzeitanalyse des modifizierten generischen Kürzeste-Pfade-Algorithmus A'_{gen} verwendet u.a., dass jede i-Iteration nur $O(|A_i|)$ Schritte benötigt. Schlage eine Implementierung vor, die dies bewerkstelligt.